

Especificação do Trabalho

Diego Passos

14 de junho de 2021

Introdução

Esse trabalho consiste em duas partes: uma de implementação e outra de experimentação. Na primeira parte, cada grupo deve implementar alguns protocolos de acesso ao meio estudados na disciplina sobre um simulador escrito em python fornecido em anexo a essa especificação. Na segunda parte, cada grupo deverá usar as implementações realizadas e uma implementação de exemplo fornecida nessa especificação para realizar diversos experimentos comparando os vários protocolos.

Primeira Parte: Implementação

Nesta fase, cada grupo implementará três protocolos: - o Aloha Puro; - o CSMA Não Persistente; e

- um protocolo MAC hipotético que simplesmente retransmite o pacote até que a transmissão seja bem sucedida, sem implementar nenhum mecanismo de controle.

Nesse terceiro protocolo, quando um pacote é recebido pela camada de enlace, ele é sempre imediatamente transmitido pelo meio físico. Se a transmissão é mal-sucedida, o protocolo hipotético simplesmente tenta uma retransmissão imediatamente. Ou seja, não são usados mecanismos para controlar efetivamente o acesso ao meio.

Já os protocolos Aloha Puro e CSMA Não Persistente devem ser implementados conforme descritos nos materiais didáticos usados na disciplina.

O Simulador

As implementações devem ser realizadas sobre um código base fornecido em conjunto com essa especificação, denominado `simulador_Template.py`. Esse código implementa um simulador de um enlace tipo barramento no qual se conectam vários nós: um receptor e um número configurável de transmissores. Na simulação, os transmissores geram pacotes a uma determinada taxa que são transmitidos para o receptor. Se há mais de um transmissor na simulação, esses competem pelo uso do barramento compartilhado, o que pode levar a colisões.

O simulador suporta uma série de opções que alteram o cenário de simulação. Quando executado com a opção `-h`, uma ajuda é exibida, resumizando todas as opções disponíveis:

```
$ python simulador_Template.py -h
```

```
Uso: simulador_Template.py [-h] [-r R] [-s S] [-i I] [-t T] [pos [pos ...]]
```

Simulador de protocolo MAC.

Argumentos posicionais:

| | |
|-----|---|
| pos | posição do nó no barramento (m). Primeiro nó é o receptor. (default: [50, 0, 100]) |
|-----|---|

Argumentos opcionais:

`-h, --help` exibe essa ajuda e sai

```

-r R      taxa de transmissão das interfaces (b/s) (default: 100000.0)
-s S      tamanho dos pacotes (B) (default: 1500)
-i I      intervalo de geração de pacotes para cada transmissor (s)
          (default: 1)
-t T      duração da simulação (s) (default: 10)

```

Mais concretamente:

- A opção `-t` recebe como argumento um valor real denotando a duração máxima da simulação em segundos.
- A opção `-i` recebe como argumento um valor real denotando o intervalo médio em segundos entre os pacotes gerados para cada transmissor.
- A opção `-s` recebe como argumento um valor inteiro denotando o tamanho em bytes dos pacotes transmitidos por cada transmissor.
- A opção `-r` recebe como argumento um valor real denotando a taxa de transmissão das interfaces dos transmissores em b/s.

Além dessas opções, os últimos argumentos denotam as posições de cada nó simulado no barramento. As posições são dadas em metros e denotam a distância (ao longo do barramento) entre o ponto de conexão de cada nó e uma das extremidades do barramento. Por exemplo, uma posição 0 denota que o nó correspondente se conecta exatamente na extremidade do barramento.

Esses últimos argumentos são uma lista das posições dos nós, sendo que o primeiro nó da lista sempre denota o receptor (os demais são sempre transmissores). Repare que o tamanho da lista determina o número de nós simulados. Por exemplo, por padrão (i.e., se nenhuma posição for especificada), o programa considera a lista “50 0 100”. Isso significa que serão simulados três nós. O receptor está a 50 metros da extremidade do barramento; um transmissor está na posição 0 (exatamente na extremidade do barramento); e o outro está na posição 100 (ou seja, a 50 metros do receptor).

Como saída, o simulador imprime a sequência de eventos relevantes ocorridos no enlace simulado. Dentre esses eventos, estão a execução de backoffs, a transmissão de pacotes pelo meio físico, a recepção bem sucedida de pacotes no receptor e a ocorrência de colisões. Além disso, ao final da simulação, o programa imprime um resumo de estatísticas de desempenho relevantes do ponto de vista do nó transmissor.

O simulador é escrito em Python 2 e foi testado com a versão 2.7.18 no Linux. Entretanto, ele deve funcionar com qualquer instalação do Python 2.7 e não utiliza nenhum pacote externo à biblioteca padrão. Versões alternativas para Python 3 também se encontram em anexo (sufixo “_python3.py”). De forma similar às versões para Python 2, elas foram testadas com a Python 3.8.5 no Linux, mas devem funcionar com qualquer instalação padrão do Python 3.8. Cada grupo pode escolher livremente se deseja trabalhar com as versões Python 2 ou Python 3.

O Template e o que Deve Ser Implementado

O template fornecido nessa especificação implementa praticamente todas as funcionalidades do simulador. O trabalho de implementação de cada grupo está restrito a apenas 5 pontos:

1. Implementar o corpo do método `MACInicializa` da classe `Interface`: esse método é chamado toda vez que um novo pacote chega para transmissão pela interface. Ele é o ponto de entrada do algoritmo do protocolo de acesso ao meio. Nele, o protocolo deve decidir como agir inicialmente com o pacote (e.g., tentar transmitir imediatamente? esperar algum tempo? verificar alguma condição do meio de transmissão?).
2. Implementar o corpo do método `MACFinalizaBackoff` da classe `Interface`: esse método é chamado toda vez que um período de backoff realizado pelo protocolo de acesso ao meio termina. Assim, esse método deve realizar os passos executados pelo protocolo imediatamente após um backoff (e.g., tentar transmitir imediatamente? verificar alguma condição do meio de transmissão?).
3. Implementar o corpo do método `MACFinalizaTentativaDeTransmissaoMalSucedida` da classe `Interface`: esse método é chamado toda vez que uma tentativa de transmissão de um pacote falha por conta de uma colisão. Aqui, o protocolo deve decidir o que fazer após uma colisão. Geralmente, isso envolve uma retransmissão, mas, para alguns protocolos, pode haver outros passos a serem realizados antes da retransmissão em si.
4. Implementar o corpo do método `MACFinalizaTentativaDeTransmissaoBemSucedida` da classe `Interface`: esse método é chamado toda vez que uma tentativa de transmissão de um pacote é completada com sucesso (ou seja, o receptor recebeu o pacote sem colisão). Aqui, tem-se a oportunidade de implementar alguma tarefa necessária ao protocolo de acesso ao meio após a conclusão da transmissão.

5. Declarar os valores de parâmetros que porventura o protocolo de acesso ao meio utilize (e.g., tamanho de janelas de backoff).

Os pontos a serem modificados estão marcados com comentários no código-fonte do template (em particular, marcados com o termo “TODO”). Os comentários providos nesses pontos dão, também, detalhes sobre os parâmetros que cada método recebe e citam métodos auxiliares que podem/devem ser usados na implementação dos quatro métodos supracitados.

Nenhuma outra modificação é necessária para implementar os protocolos pedidos nessa especificação. Em especial, solicita-se que os grupos evitem ao máximo alterar outras partes do código.

Além do template, encontra-se anexo a essa especificação o arquivo `simulador_AlohaSlotted.py`. Trata-se de uma implementação do protocolo Slotted Aloha feita sobre o template, ilustrando como o mesmo deve ser utilizado.

Segunda Parte: Experimentação

Ao final da primeira parte do trabalho, cada grupo deve ter (além do template) quatro protocolos implementados:

1. o Slotted Aloha (fornecido como exemplo);
2. o Aloha Puro (implementado pelo grupo);
3. o CSMA Não Persistente (implementado pelo grupo); e
4. o protocolo hipotético que não contém mecanismos de controle de acesso ao meio (implementado pelo grupo).

Utilizando essas implementações, cada grupo deverá executar uma série de experimentos, analisando seus resultados. Mais especificamente, cada grupo deverá realizar, analisar e documentar os seis experimentos detalhados a seguir.

Experimento 1

Nesse experimento, cada grupo deverá executar simulações com todos os quatro protocolos. Todas as simulações devem ter duração de 1000 segundos (especificado através do parâmetro `-t`). As simulações terão apenas dois nós a 100 metros de distância um do outro. Para cada protocolo, deverão ser realizadas simulações com intervalos de geração de pacotes (opção `-i`) de 2, 1, 0,5, 0,1 e 0,05 segundos. Os demais parâmetros da simulação deverão ser deixados nos valores padrão do simulador.

Com base nos resultados, cada grupo deverá responder às seguintes perguntas:

1. Algum método foi melhor que os demais?
2. Algum método foi pior que os demais?
3. Se sim para alguma das duas perguntas acima, por quê? Algum parâmetro dos métodos afeta o desempenho nesse cenário?

Experimento 2

Esse experimento é idêntico ao anterior, exceto pelo fato de que a distância entre os dois nós simulados deve ser configurada para 100000 metros, ao invés dos 100 metros usados anteriormente.

Com base nos resultados desse experimento e do experimento anterior, o grupo deverá responder:

1. O aumento da distância dos nós causou algum prejuízo significativo de desempenho?

Experimento 3

Esse experimento é similar ao Experimento 1, mas agora com três nós (dois transmissores e um receptor). Mais especificamente, cada grupo deverá executar simulações com todos os quatro protocolos. Todas as simulações devem ter duração de 1000 segundos (especificado através do parâmetro `-t`). O nó receptor deverá ser colocado na posição 50 metros, enquanto um transmissor deverá ser posicionado em 0 metros e o outro em 100 metros (de forma que ambos os transmissores estejam a 50 metros do receptor). Para cada protocolo, deverão ser realizadas simulações com intervalos de geração de pacotes (opção `-i`) de 2, 1, 0,5, 0,1 e 0,05 segundos. Os demais parâmetros da simulação deverão ser deixados nos valores padrão do simulador.

Com base nos resultados, cada grupo deverá responder às seguintes perguntas:

1. Algum método foi melhor que os demais?

2. Algum método foi pior que os demais?
3. Se sim para alguma das duas perguntas acima, por quê? Algum parâmetro dos métodos afeta o desempenho nesse cenário?

Experimento 4

Para esse experimento, deverão ser usados apenas o CSMA Não Persistente, o Aloha Puro e o Slotted Aloha. Deverão ser executadas simulações de 1000 segundos de duração com intervalo de geração de pacotes fixo em 0,1 segundos. Em todas as simulações, o receptor deverá ser posicionado em 0 metros. Por outro lado, deve-se variar o número de transmissores de 1 a 6. Independentemente do número de transmissores, todos devem ser colocados na posição 100 metros do barramento. Os demais parâmetros da simulação deverão ser deixados nos valores padrão do simulador.

Com base nos resultados, cada grupo deverá responder às seguintes perguntas:

1. Algum método foi melhor que os demais?
2. Algum método foi pior que os demais?
3. Se sim para alguma das duas perguntas acima, por quê? Algum parâmetro dos métodos afeta o desempenho nesse cenário?

Experimento 5

Ao contrário dos experimentos anteriores, nesse cada grupo deverá projetar o experimento (e.g., quantos nós? posicionados onde? qual intervalo de transmissão de pacotes?). O objetivo é avaliar como a assimetria afeta cada protocolo. Aqui, “assimetria” se refere à diferença entre as distâncias dos transmissores em relação ao receptor. Note que em todos os experimentos anteriores, todos os transmissores estavam sempre a uma mesma distância do receptor.

O que se deseja descobrir aqui é se o desempenho geral ou de cada transmissor individual é afetado em cada protocolo pelo fato de transmissores estarem a distâncias diferentes. Essa pergunta deve ser respondida com base nos resultados das simulações executadas. Além disso, se a assimetria afeta algum dos protocolos, cada grupo deve explicar por que isso ocorre.

Experimento 6

Assim como no experimento anterior, esse último experimento deve ser projetado pelo grupo. Nesse caso, o objetivo é entender como cada protocolo é afetado pelo aumento da distância entre transmissores e receptor **quando há mais de um transmissor**. Nesse experimento, deve-se considerar um cenário simétrico, isto é, todos os transmissores estão a uma mesma distância do receptor.

Com base nos resultados das simulações, cada grupo deverá analisar o efeito do aumento da distância no desempenho dos protocolos. Se a distância influenciar no desempenho de algum protocolo, o grupo deverá explicar o por quê desse efeito.

Observação: para esse experimento, sugere-se que sejam avaliadas distâncias na casa de milhares de quilômetros.

Relatório

Como parte do trabalho, cada grupo deverá escrever um relatório sucinto documentando o que foi realizado. Em particular, o relatório deverá:

1. Explicar como o grupo implementou cada um dos protocolos solitados.
2. Detalhar a metodologia utilizada nos Experimentos 5 e 6.
3. Exibir (na forma de tabelas e/ou gráficos) os resultados obtidos em cada experimento, discutindo-os com foco especial em responder às perguntas indicadas nessa especificação.

Não há limites superiores ou inferiores de páginas para esse relatório, nem regras específicas de formatação. Apenas como um guia geral, 7 páginas devem ser suficientes (embora não necessárias) para contemplar todas as informações pedidas.

Critério de Avaliação

Os trabalhos serão avaliados em uma escala de 0 a 10. A pontuação será distribuída da seguinte maneira:

- Até 0,5 ponto para a implementação do protocolo hipotético sem mecanismos de controle de acesso ao meio.
- Até 2,0 pontos para a implementação do Aloha Puro.
- Até 2,5 pontos para a implementação do CSMA Não Persistente.
- Até 0,5 ponto para a execução e análise do Experimento 1.- Até 0,5 ponto para a execução e análise do Experimento 2.
- Até 0,5 ponto para a execução e análise do Experimento 3.
- Até 0,5 ponto para a execução e análise do Experimento 4.
- Até 1,0 ponto para o projeto, a execução e a análise do Experimento 5.
- Até 1,0 ponto para o projeto, a execução e a análise do Experimento 6.
- Até 1,0 ponto para a qualidade do relatório.

Composição dos Grupos e Entrega

Os grupos poderão ser formados com no máximo 5 e no mínimo 2 alunos.

Os trabalhos devem ser entregues através dessa atividade do Google Classroom por apenas um dos integrantes do grupo. A entrega deve ser feita em formato zip, contendo o relatório em PDF e as implementações realizadas. O relatório deve listar claramente os integrantes do grupo. Após a entrega, **não serão aceitas modificações na lista de integrantes**.

Os trabalhos serão aceitos sem penalização até a data limite especificada nessa atividade do classroom. Trabalhos entregues após essa data serão aceitos, mas com uma penalidade de 1 ponto por dia de atraso (até o limite de 10 dias de atraso).